IN THE SPECIFICATION

Please delete the paragraph on page 8 beginning with line 1 and ending with line 12 and replace with the following:

> In a request phase of arbitration process (Fig. 2(a)), each first portion, which receives requests generated by corresponding ingress ports, generates request signals. Each request signal relates to a connection to be made between that ingress port and one of the egress ports, and is transmitted to the corresponding one of the ~~sfeond~~ second portions 31, 33, 35, 37 along a respective request signal path. In each request phase, the requests generated by the first portions are derived from all of the requests generated by the ingress ports which have not yet been satisfied (in other words the first portions 21, 23, 25, 27 may keep a store of such requests, and generate its own requests based on this store), but taking into account any constraints on the arbitration unit. The detail of how this is done are known to an expert, and will not be considered here.

Please delete the paragraph on page 9 beginning with line 4 and ending with line 15 and replace with the following:

> For ease of understanding, each of the locations (such as 270, 271, [[271]] 272, 273) on each of the first portions is marked with an integer 0, 1, 2, 3 indicating the pointer sequence 93 is followed by 0). Each of the grant pointer locations (such as 310, 311, 312, 313) on each of the first and second portions is marked with an integer 0, 1, 2, 3 indicating their grant pointer sequence, that is the value M. M=3 is followed by M=0. Also, 4-element columns (such as 317) are added before the second portions 31, 33, 35, 37 to indicate schematically the values of M* for that

pointer. For example, column 317 has elements 0, 1, 3, 2 to indicate that, as the pointer of the second portion 31 moves ~~though~~ through the successive locations 310, 311, 312 and [[312]]313, it points successively at the first portions 21, 23, 27, 25 which correspond to the respective ingress ports 0, 1, 3, 2.

Please delete the paragraph on page 9 beginning with line 17 and ending with line 23 and replace with the following:

[[dhe]]The result [[if]]is as shown in Fig. [[3(b*]]3(b), which gives for each [[pf]]of the egress ports P a column showing for any "original pointer value" M, the "mapped pointer calue" M* under Eqn. (1) (i.e. the next ingress port to which the location points in the ~~present~~ present invention). For example, the column under P=0 (the egress port corresponding to second portion 31) shows that as the location 310, 311, 312, 313[[, 314]] moves through positions M=0, M=1, M=2, and M=3, M* moves through the sequence 0, 1, 3, 2.

Please delete the paragraph on page 11 beginning with line 1 and ending with line 8 and replace with the following:

switch cycle (equal to the vector per tensor size), where the maximum possible number of connections are made every cycle to satisfy a number of pre-ordained connections (which are not considered in detail in this document) and queued unicast/multicast/broadcast requests. The process is pipelined in order hide the ~~considfrable~~ considerable processing required to generate each connecting vector. The fourth embodiment has the overall structure shown in Fig. 1, but the operation of the arbitration unit differs from that of known systems as discussed below.

Please delete the paragraph on page 12 beginning with line 16 and ending with line 29 and replace with the following:

As explained above in relation to the other embodiments of the present invention, a separate Hashed Weighted Round Robin is used by each egress 'port' to select one of the incoming requests to grant. For each round robin, a pointer indicates the last ~~ingrfss~~ ingress port to which a grant was issued which was subsequently accepted. When a new set of requests arrives, the ingress ports are tested in a "hashed" order, that is one according to the present invention, for example defined by Eqn. 1, starting from the one after that indicated by the pointer, until the first with an unmasked request is found: this request is granted. The hashed order of the ports is different for each of the round robins to avoid pointer synchronisation effects. Each round robin also maintains a set of weight registers, one weight per ingress port. A request is considered masked (i.e. not a candidate for selection in the round robin) if its weight is zero: this represents a connection that is exceeding its bandwidth allocation. If the round robin finds all active request are masked